

1994018761

N94-23234

1993 NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

JOHN F. KENNEDY SPACE CENTER
UNIVERSITY OF CENTRAL FLORIDA

54-66

197190

P 14

A FRAMEWORK FOR SFC/DC SYSTEM DATA ANALYSIS

PREPARED BY:

Dr. Martha A. Centeno

ACADEMIC RANK:

Assistant Professor

UNIVERSITY AND DEPARTMENT:

FAMU/FSU College of Engineering
Department of Industrial Engineering

NASA/KSC

DIVISION:

Shuttle Operations

BRANCH:

Vehicle Processing

NASA COLLEAGUE:

Amanda Mitskevich

DATE:

August 26, 1993

CONTRACT NUMBER:

University of Central Florida
NASA-NGT-60002 Supplement: 11

A FRAMEWORK FOR SFC/DC SYSTEM DATA ANALYSIS

Martha A. Centeno¹

Athena Colucci

Department of Industrial Engineering
FAMU/FSU College of Engineering
Tallahassee, Florida 32316

ABSTRACT

The Shop Floor Control / Data Collection (SFC/DC) system is one of four major modules of the Integrated Work Control System being implemented at the NASA Kennedy Space Center. The SFC/DC system, which has been in operation approximately one and a half years, collects detailed information regarding the duration of tasks as well as work stoppages duration. This information needs to be effectively analyzed, so as to be able to manage the shuttle processing activities under the continuous improvement philosophy. We have studied the contents and form of the SFC/DC database, and based on the results of such study, we have designed a Shop floor Modeling, Analysis, and Reporting Tool (SMART). SMART is a collection of compatible tools (off-the-shelf as well as customized C-based tools) that work cohesively to prepare inputs for the analysis models and process the various models. The feasibility of SMART has been demonstrated through an initial prototype.

1. INTRODUCTION

This report is a summary of two extended reports that were written for the NASA TP-OAO office of the Kennedy Space Center. A multitude of valuable, yet confidential, information has been suppressed from this report.

This effort sought to design a cohesive framework for the effective analysis of the data being collected through the Shop Floor Control / Data Collection (SFC/DC) system. The design of such framework required a thorough understanding of the contents and form of the data in the SFC/DC database. Once this was established, a conceptual design and data exchange protocols for a Shop Floor Modeling, Analysis, and Reporting Tool (SMART) were devised. The SMART framework incorporates methods and tools to answer questions regarding delay duration such as:

- Do delays follow a pattern within a flow?
- Are delays correlated to the orbiter systems?
- Are delays correlated to the shuttle elements?
- Are delays correlated to the shuttle processing facility?
- What delays cause the most impact to the whole process?

Two protocols to provide inputs to the analysis models were considered: *directly from the*

¹ This effort has been funded through a NASA/ASEE fellowship and through a grant from the Council on Research and Creativity of the Florida State University.

database and from intermediate text files. At the moment of writing this report, NASA engineers had limited access to the various SFC relational tables; therefore, we chose the second protocol: *intermediate text files*. The procedures required to work with these models include instructions to prepare the text files and to transform them into the format of the analysis tool. STATGRAPHICS® is being recommended as the analysis tool based on its ease of use and its *macros* programming capabilities.

Section 2 of this report describes briefly what SFC is and how its database is populated. Section 3 describes the conceptual design for SMART. Section 4 gives a sample a proposed procedures to analyze SFC data. Section 5 summarizes the results of this effort and their significance to NASA and ASEE.

2. SFC/DC AND ITS DATABASE

SFC is one of the four clusters that make up the Integrated Work Control System (IWCS), which will integrate the shuttle processing databases at Kennedy Space Center (KSC). The IWCS framework will enable communication among the four clusters and add new data collection protocols (Figure 1).

The Shop Floor Control (SFC) module has been operational for one and a half years; however, at this stage, automatic links to the other 3 modules have not been implemented yet; except for a partial link to IOS (CASPR). SFC revolves around a DB/2 database with PFORMS acting as the database management system (DBMS). Tables in the SFC database are populated either from the IOS database or directly by the user. The main dynamic entity in the SFC and IOS database is a *task*; thus, the physical storage location and update privileges are driven by the status of the WAD. Figure 2 shows the various *states* that a WAD may experience throughout its life-cycle as well as the software systems (SFC or IOS) governing the WAD at each one of these statuses.

The majority of updates to WADs are done by technicians. Once all tasks in IOS that fall in the 11-day window are transferred to SFC, they are *available* to the technicians to work on. (table *orderopn*). When technicians are ready to work on a task, or when they want to change a task's status to hold, they clock in and provide pertinent information. This information does not go directly into *tsf01_delay* or *tsf02_task_worked*; rather, it goes to other SFC tables, such as ACTVEMPL from which the information for *tsf01_delay* and *tsf02_task_worked* is extracted once a week, as shown in Figure 3.

Tables in the SFC database are populated either from the IOS database or directly by the user. Figure 4 depicts the data exchange protocol that is being utilized to link IOS and SFC databases. This protocol is executed in batch mode, and it enables *almost real time* updates to both databases. It falls short of real time because the protocol is executed either once a day (from IOS to SFC) or twice a day (from SFC to IOS). IOS is not a DB/2 database; consequently, when transferring data to (and from) SFC, it is necessary to *extract* the data to a format neutral to both databases (ASCII or binary) and then *load* the data into the database unique format. This conversion is done automatically by a series of COBOL and REXX/SQL procedures. These procedures have been designed to preserve data integrity: they avoid duplicate records, prevent unauthorized updates, create tables for unexpected errors in the data, and so forth.

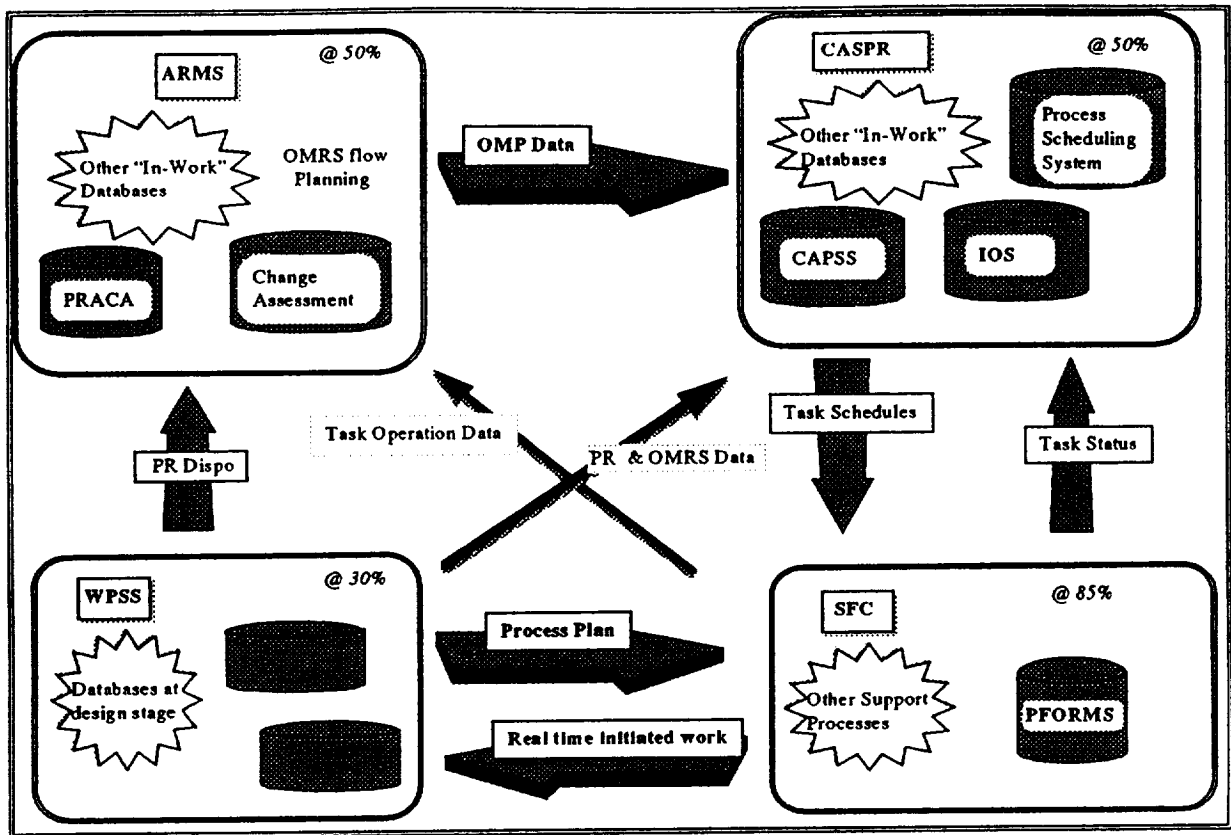


Figure 1: The IWCS Concept

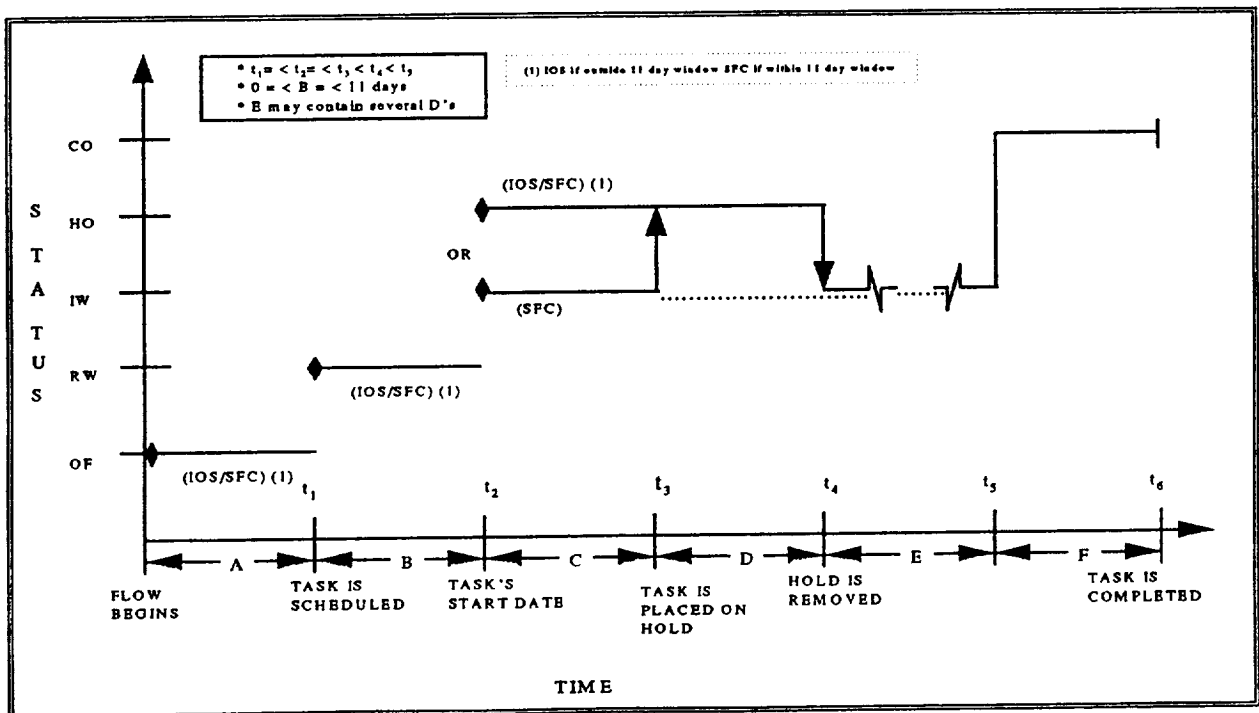


Figure 2: Generic Realization of a WAD

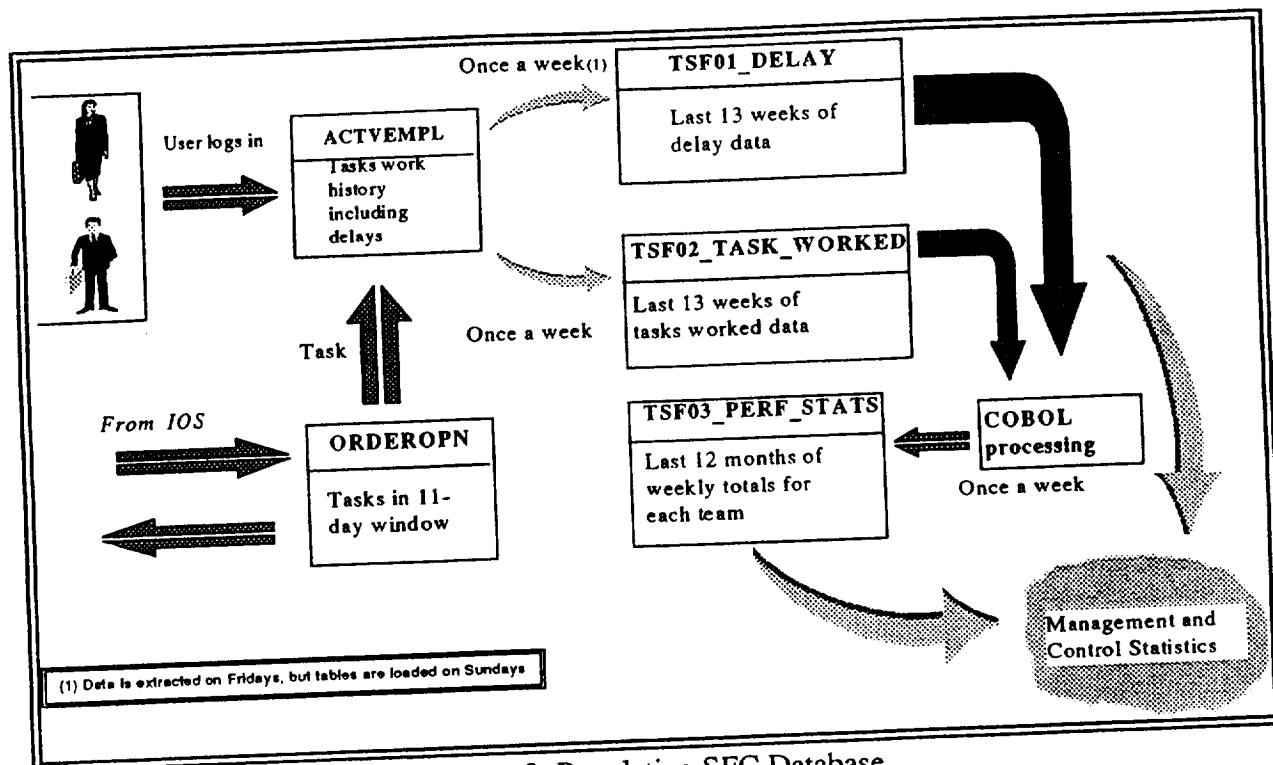


Figure 3: Populating SFC Database

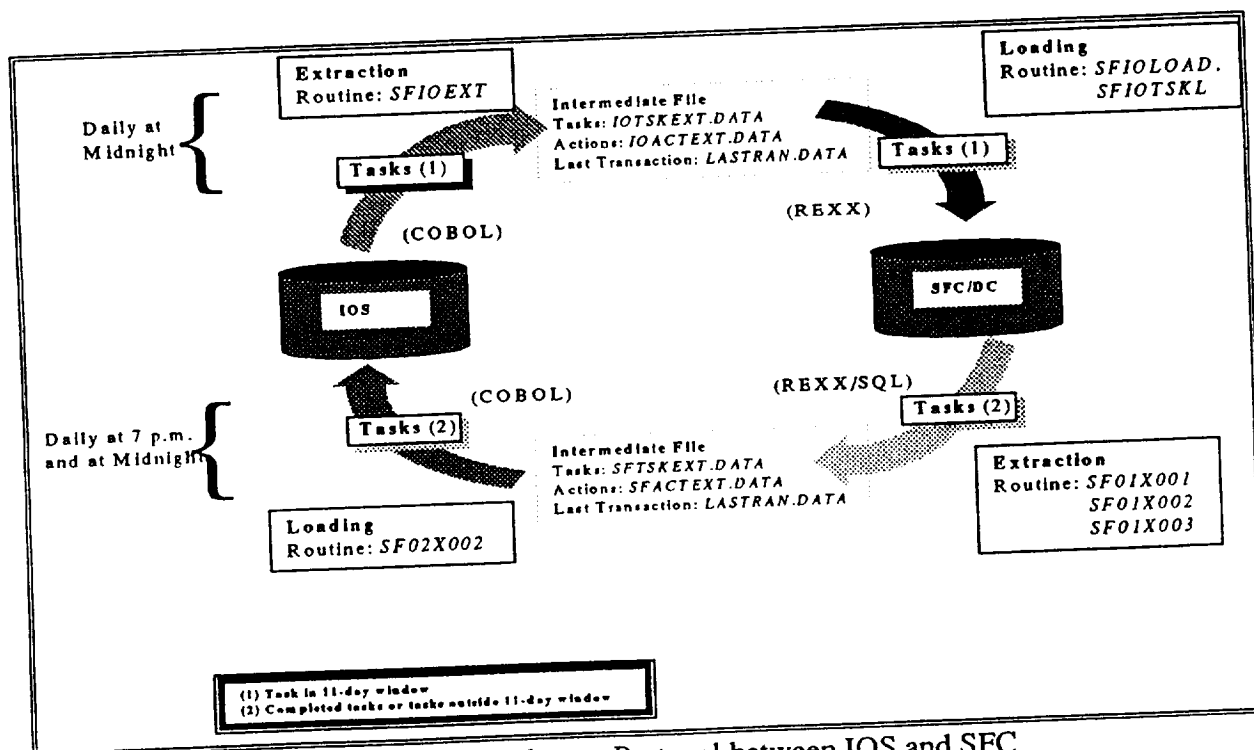


Figure 4: Data Exchange Protocol between IOS and SFC

The SFC database is truly a DB/2 database; therefore, it is accessible through DB/2 SQL commands and QMF SQL, i.e. it is not necessary to go through the PFORMS application to extract data out of it. Figure 5 shows the various ways in which NASA engineers may gain direct access to the SFC/DC database, with QMF being the most suitable via to achieve the goals of this on-going effort.

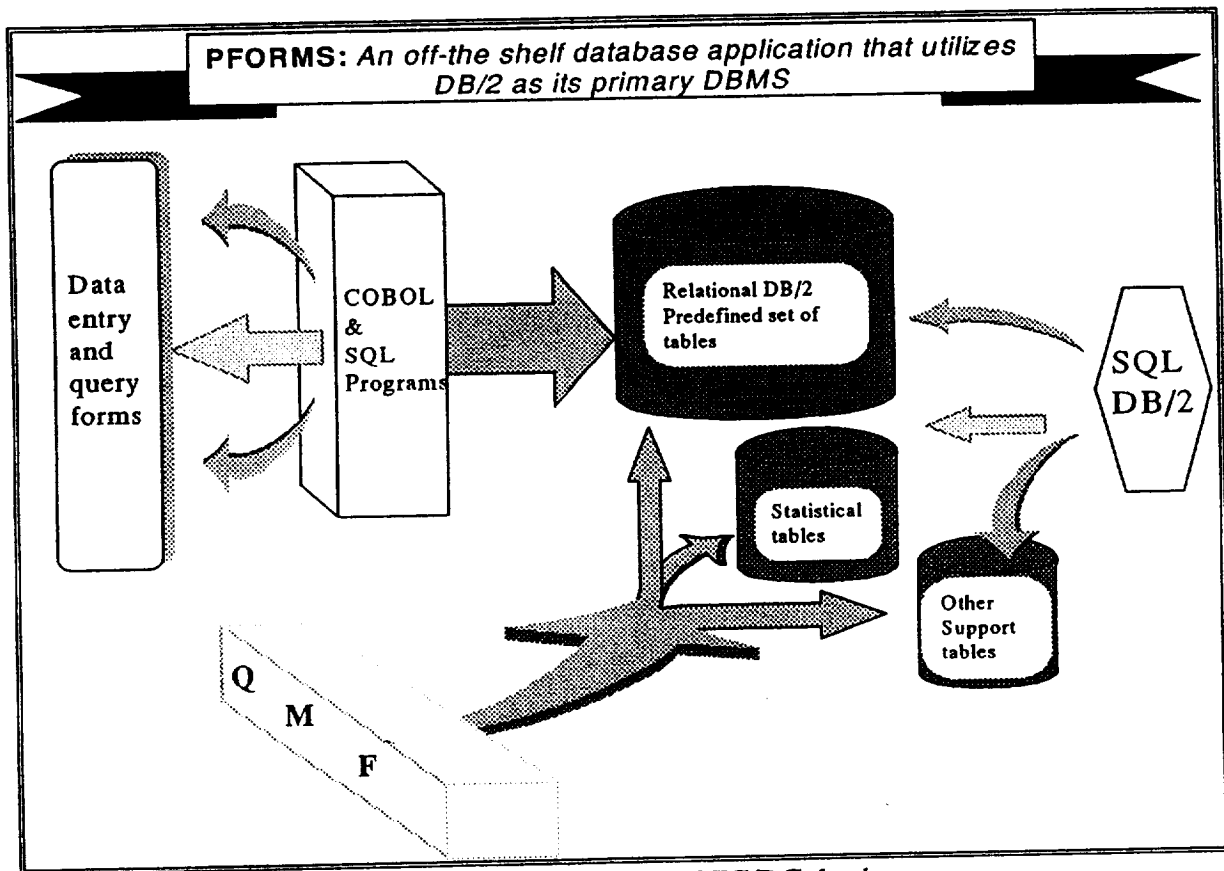


Figure 5: Gaining access to SFC/DC database

In a final report given to the TP-OAO office, we have given more detailed information regarding the populating of the SFC/DC database, as well as extensive documentation on the various tables in it. The reader is directed to contact TP-OAO office if interested in more details than those provided in this report.

3. THE SMART CONCEPT

The Shop floor Modeling, Analysis and Reporting Tool (SMART) is a framework designed around a relational database (Figure 6). The database contains statistical information regarding work time and delay duration. Under this framework, a link to the SFC/DC database (via text files) has been set. The framework is sensitive to the user of it. We have identified three possible users:

- * The *Statistical user* interacts with SMART to execute pre-defined analysis procedures. This user would request from the *designer user* modifications to SMART, so that new analysis models can become part of it.
- * The *Modeling user* interacts with SMART to extract raw data as well as statistical summaries from its database. This user would request from the *designer user* modifications to SMART, so that new modeling ideas can be readily tested.
- * The *Designer user* interacts with SMART through its SQL query language and other DBMS tools. This user would service the *statistical user* and *modeling user* needs regarding the modeling and analysis activities at the TP-OAO office.

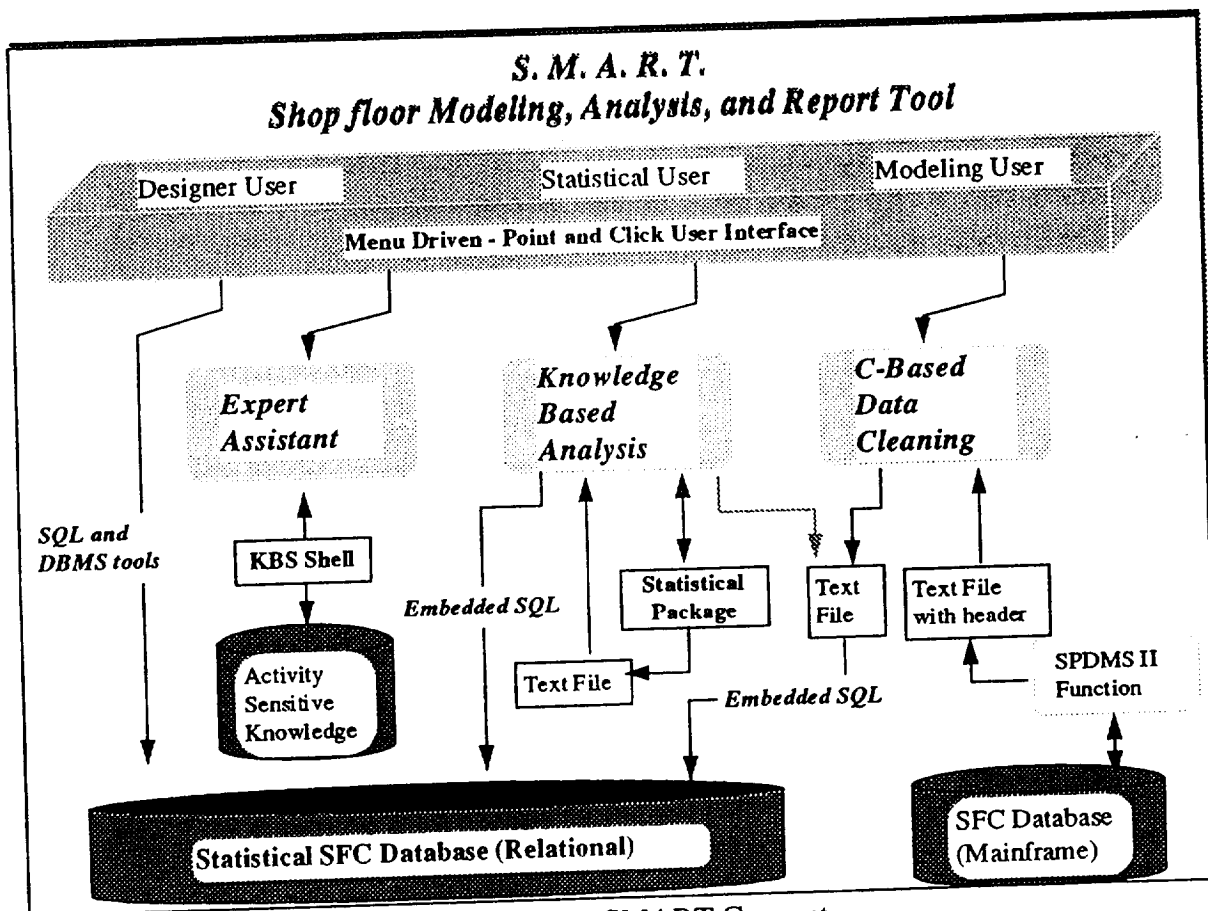


Figure 6: The SMART Concept

From Figure 6, we can see that there three major components to SMART, plus the access for the *designer user*. Each one of these components adds capabilities to SMART as follows:

- * SQL and DBMS Tools Access: It enables the creation of tables as new analysis

requirements are defined. It enhances accessibility of data from the analysis point of view. It enables the creation of "canned" reports using the **report** tool of the DBMS. It also enables the development of data entry forms as well as of specific applications (as needed) using the **forms** and **menu builder** tools.

- * **Expert Assistant:** It is an knowledge based expert system that operates in three different modes. It can be a tutor regarding the various statistical analysis, or it can be an assistant to the NASA engineer in choosing appropriate analysis methods, or it can provide user-sensitive and procedure sensitive help for the entire SMART framework.
- * **Knowledge Based Analysis:** This component computes descriptive statistics and estimates on delay duration, conducts various correlation analyses, conduct various hypotheses testing on the similarity of facilities, computes various joint probability functions for delay duration, and it enables the collection of historic behavior for possible use in other modeling activities.
- * **C-based Data Cleaning** It enables a semi automated link between the statistical database and current SFC database by preparing extraction procedures and then cleaning the resulting files. Once the files are cleaned, it utilizes embedded SQL to insert the data into the statistical tables. It also enables the creation of input files based on the format of the analysis and modeling tool to use.

The feasibility of SMART has been shown through a small prototype. This prototype includes a menu driven interface, through which a cleaning procedure can be executed. The data extraction and data cleaning prototype is partly operational for analyses on *total delayed time for a given flow*, *weekly delay duration*, and *probability functions for delay duration*. In the next section, we give one of such analysis procedures.

4. A SAMPLE PROCEDURE

The procedures developed are based on the premises that data in the SFC database is reliable, and that necessary software routines will be developed to achieve the objectives of the data extraction specifications.

Some variables definitions in this section utilized standard database notation which calls for the prefixing of *fields* with the *table* name. For example, in the definition below, **actvempl** is the name of the table that contains the field *sdate* (start date of work), *stime*, *actcdate*, and *actctime*. The + sign indicates that the two fields will be concatenated. Boldface font is being used as a reminder of the notation.

$$W_{ijk} = \text{actvempl.sdate} + \text{actvempl.stime} - \text{actvempl.actcdate} + \text{actvempl.actctime}$$

All the field names used in the definitions correspond to the actual names in the physical SFC table. The meaning of each field can be found in the final report given to the TP-OAO office.

Most procedures described herein are composed of 4 major steps:

1. Transfer of summary information from SFC data tables to statistical tables residing in the mainframe. This transfer is done automatically by issuing SQL statements with the INSERT/SELECT combination.
2. Extraction of summary information from statistical tables to text files, so as to generate reports and/or uses STATGRAPHICS for analysis. This is done by executing SQL statement SELECT and utilizing QMF extraction capabilities in conjunction with OS/2 data transfer protocol.
3. Cleaning of data text files. When extracted the files have some header information that must be removed before STATGRAPHICS can use the file. This cleaning is done by running a C program.
4. Analysis of data using STATGRAPHICS, which includes the generation of appropriate graphs and summaries.

Some of the queries given in this report have been tested using the QMF version of SQL; others have not been tested; however, we believe that no major changes are needed to these queries to make them work as desired.

The nature to of the QMF version of SQL has forced us to extract fields that may seem redundant from the analysis point of view, but these *fields* and *grouping/column functions* are necessary to make QMF SQL work.

Total delay time for a flow

OBJECTIVES: To compute (for an entire flow) 1) the total delayed time, 2) the total delayed time per task, 3) the total delayed time per delay category

DEFINITIONS: A *delay* refers to the amount of time (hours) that technicians could not work on a task due to the lack of the appropriate resources (materials and personnel).

$$D_{ijk} = tsf01_delay.delay_duration$$

$$TDS_{ij} = \sum_{k=1}^{ND_{ij}} D_{ijk}$$

$$TDT_i = \sum_{j=1}^{NS_i} TDS_{ij} = \sum_{j=1}^{NS_i} \sum_{k=1}^{ND_{ij}} D_{ijk}$$

$$TDF = \sum_{i=1}^{NT} TDT_i = \sum_{i=1}^{NT} \sum_{j=1}^{NS_i} \sum_{k=1}^{ND_{ij}} D_{ijk}$$

$$TDCF_l = \sum_{i=1}^{NT} \sum_{j=1}^{NS_i} \sum_{k=1}^{ND_{ij}} D_{ijk} \quad \text{for each } D_{ijk} \text{ with delay code } = l$$

where:

- D_{ijk} = k th delay time duration (hours) for the i th task at the j th shop department, regardless of code type.
- TDS_{ij} = Total delay time for the i th task at the j th shop department
- TDT_i = Total delay time for the i th task
- TDF = Total delay time for the flow
- $TDCF_l$ = Total delay time for code l in the entire flow
- ND_{ij} = number of delays, regardless of code type, that the i th task experienced at the j th shop department. $ND_{ij} = 1, 2, 3, \dots$
- NS_i = number of shops that worked on the i th task. $NS_i = 1, 2, 3, \dots$
- NT = number of executed tasks in the flow. $NT = 1, 2, 3, \dots$

WHEN TO EXECUTE PROCEDURE: The TDF, TDF, and TDT should each be computed at the end of a flow. The TDF report must be generated and trend analyses must follow, so as to identify any adjustments necessary in forthcoming flows.

DATA EXTRACTION and ADDITION:

1. A text file (*tdf.dat*) which contains **sts_num**, **orbiter_no**, **NT**, and **TDF** (query 4.2.7)
2. A text file (*tdt.dat*) which contains actual entries for each task in a flow (query 4.2.8)
3. A text file (*tdcf.dat*) which contains total delay time for delay code for the given flow. (query 4.2.9)
4. 1 record to **total_delay_flow** table (query 4.2.1)
5. **NT** records to **total_delay_task_flow** table (query 4.2.3)
6. Up to 84 records to **total_delay_code_flow** table (query 4.2.5)

PROCEDURE:

1. Ask user for **sts_no**, **orbiter_no**
2. Create an ASCII file (*before.sql*) with the following SQL commands. The lines numbered 4.1.i are not part of *before.sql*; they are comments in this report to explain what each SQL command is supposed to achieve.

4.2.1 Command to determine the total delay time in the given flow. It produces 1 record with 3 fields (out of 4 in the target schema), and it inserts the

record into the table total_delay_flow

```
INSERT INTO total_delay_flow (sts_no, tdf, num_of_delays)
SELECT      sts_no, orbiter_no, sum(delay_duration), count(*)
FROM        historic_tsfl_delay
WHERE       sts_no = (id for a flow, e.g. sts-51)
```

- 4.2.2 *Command to update the record created by query 4.2.1. This query fills the 4th field in the target schema.*

```
UPDATE total_delay_flow
SET   orbiter_no = (orbiter use in this sts_no. Given by the user)
WHERE sts_no = (id for a flow, e.g. sts-51)
```

- 4.2.3 *Command to find out the total delay time per task in the given flow. It produces one record per task executed in the given flow. Each resulting record has 6 fields (out of seven in the target schema). These records are inserted into the total_delay_task_flow*

```
INSERT INTO total_delay_task_flow (sts_no, task_no, wad, wad_type,
                                   tdt, num_delays, avg_del, std_del, max_del, min_del)
SELECT      sts_no, task_no, max(wad), max(wad_type),
            sum(delay_duration), count(*), avg(delay_duration),
            std(delay_duration), max(delay_duration),
            min(delay_duration)
FROM        historic_tsfl_delay
WHERE       sts_no = (id for a flow, e.g. sts-51)
GROUP BY    sts_no, task_no
```

- 4.2.4 *Command to update the records created by query 4.2.3. This query fills the 7th field in the target schema.*

```
UPDATE total_delay_task_flow
SET   orbiter_no = (orbiter use in this sts_no. Given by the user)
WHERE sts_no = (id for a flow, e.g. sts-51)
```

- 4.2.5 *Command to find the total delay time per delay code in the given flow. This query creates up to 84 records since there are 84 different codes. Each record has 4 fields (out of 5 in the target schema). The records are inserted into the total_delay_code_flow table*

```
INSERT INTO total_delay_code_flow (sts_no, delay_cat, tdt,
                                   occurrence, avg_del, std_del, max_del, min_del)
SELECT      sts_no, delay_cat, sum(delay_duration), count(*),
            avg(delay_duration), std(delay_duration),
```

```

max(delay_duration), min(delay_duration)
FROM      tsf01_delay
WHERE      sts_no = (id for a flow, e.g. sts-51)
GROUP BY  sts_no, delay_cat

```

4.2.6 Command to update the records created by query 4.2.5. This query fills the 5th field in the target schema.

```

UPDATE total_delay_code_flow
SET    orbiter_no = (orbiter use in this sts_no. Given by the user)
WHERE  sts_no = (id for a flow, e.g. sts-51)

```

3. Execute SQL commands in *before.sql* (only if the process is automated)
4. Create ASCII files (names given at each query) with the outputs of the following SQL commands, using the QMF interface in conjunction with OS/2 data transfer protocol. Transfer text files to appropriate sub directory (where C program for cleaning data resides).

4.2.7 Query to extract delay time summaries for each flow (*tdf.dat*)

```

SELECT  sts_no, orbiter_no, tdf, num_of_wads
FROM    total_delay_flow

```

4.2.8 Query to extract delay time summaries for each task in the given flow (*tdt.dat*)

```

SELECT  tdt, wad, wad_type, num_delays
FROM    total_delay_task
WHERE   sts_no = (id for a flow, e.g. sts-51)
ORDER BY wad

```

4.2.9 Query to extract delay time summaries for each delay code in the given flow (*tdcf.dat*)

```

SELECT  delay_cat, tdcf, occurrence
FROM    total_delay_code_flow
WHERE   sts_no = (id for a flow, e.g. sts-51)
ORDER BY tdcf DESC

```

5. Open text files *tdf.dat*, *tdt.dat*, read only mode.
6. Create a new report file: *dtfstst##.out*
7. Count the number of records (NT) in *tdt.dat* and *tdcf.dat* (NDC)
8. Allocate arrays of size NT for *tdt*, *orbiter*, *wad*, *wad_type*, *num_delays*, of size NDC for code and *tdc*
9. Set *min_twt* = 1000000 and *max_twt* = -10000.

10. *loop1*: i = 1 to NT
 - 10.1 Read from **tdt.dat**: **orbiter(i)**, **wad(i)**, **wad_type(i)**, **tdt(i)**
 - 10.2 If **min_tdt** > **twl(i)** Then
 - min_tdt** = **tdt(i)**
 - min_wad** = **wad(i)**
 - endif
 - 10.3 If **max_tdt** < **tdt(i)** Then
 - max_tdt** = **tdt(i)**
 - max_wad** = **wad(i)**
 - endif
- end *loop1*
11. *loop2*: i=1 to NDC
 - 11.1 Read from **tdcf.dat**: **code(i)**, **tdc(i)**
- end *loop2*
12. Locate in **tdf.dat** the value of **TDF** for the given flow
13. Print report as laid out in Figure 4-4 and Figure 4-5.
14. Write a message interpreting the meaning of the values in the report
15. Run STATGRAPHICS and import text file (**tdf.dat**), by using the following instructions:
 - 15.1 click on utils, click on procedures macros, click on run macro
 - 15.2 choose **SFCITDF1** and press F6 key.

5. SUMMARY

The results reported here include the following:

1. A better understanding of the accessibility of the SFC database. We investigated issues pertaining to the physical storage of the SFC database as well as to the data base management system (DBMS) administering the SFC database. The fact that DB/2 is the DBMS used for SFC opens the possibility of accessing SFC 1) directly via SQL commands, 2) directly via a C-based program, specially designed and developed for SFC, or 3) indirectly via LSOC engineers. Each approach has its advantages and disadvantages, and we have discussed them in the TP-OAO report.
2. Specific data extraction protocols are being recommended. These protocols address data extraction from existing tables in the SFC database onto new tables as well as from the new tables onto ASCII files, so as to enable the interface with the statistical tool.

3. Specific recommendations on direct access to SFC by a NASA engineer are being made. These recommendations have been made to facilitate the communication of NASA's needs to LSOC software engineers.
4. Design of SMART, a framework that will enable NASA engineers to manipulate SFC data, analyze it, and do other modeling activities, without being concerned with the knowledge of SQL. This framework utilizes existing software to carry out the actual statistical analyses, and it includes a customized expert system to interpret the results before they are reported to NASA.
5. Specific procedures have been developed to conduct a variety of statistical analyses on the SFC data. These procedures give as much detail as necessary to enable the engineer to conduct analysis even if SMART is not implemented. Some of these procedures have been partially tested.
6. Specific subsets of SMART have been identified as candidates for an applied research proposal.
7. A worthwhile, unique experience for both Ms. Colucci and Dr. Centeno. Among other things that make this experience worthwhile are the facts that Ms. Colucci has seen first hand the application of some of the IE techniques she has been learning at school, and Dr. Centeno has accumulated new examples that can be used in courses such as Operations Research, Simulation, and Statistics. Similarly, NASA engineers have become more familiar with databases, SFC, and statistical analysis in general.

REFERENCES

- Cheu, D. & B. Linden** (1990), SOL Language Reference Manual, V6.0, ORACLE Corporation, 500 Oracle Parkway, Redwood Shores, CA 94065. (800) 633-0598.
- Coppola, A.** (1988), ORACLE Tools for MS-DOS and OS/2: Installation User's Guide, V6.0, ORACLE Corporation, 500 Oracle Parkway, Redwood Shores, CA 94065. (800) 633-0598.
- Lapin, L.L.** (1990), Probability and Statistics for Modern Engineering, Second Edition, PWS-Kent Publishing Company, Boston.
- Scheffe, H.** (1970), The Analysis of Variance, 6th printing, Wiley & Sons, Inc. Canada